



# Recognizing and Eliciting Weakly Single Crossing Profiles on Trees

Palash Dey

Indian Institute of Science, Bangalore

November 15, 2016

## Abstract

The domain of single crossing preference profiles is a widely studied domain in social choice theory. It has been generalized to the domain of single crossing preference profiles with respect to trees which inherits many desirable properties from the single crossing domain, for example, transitivity of majority relation, existence of polynomial time algorithms for finding winners of Kemeny voting rule, etc. In this paper, we consider a further *generalization* of the domain of single crossing preference profiles on trees to the domain consisting of all preference profiles which can be extended to single crossing preference profiles with respect to some tree by adding more preference orderings to it. We call this domain the *weakly single crossing domain on trees*. It is known that most of the desirable properties of the single crossing domain on trees continue to hold for the weakly single crossing domain on trees. In this paper, we prove several interesting structural properties of the weakly single crossing domain on trees and exploit them to design efficient algorithms for recognizing and eliciting weakly single crossing preference profiles on trees. In particular, we begin with showing that, for every weakly single crossing preference profile  $\mathcal{P}$  on trees, there exists a *unique* inclusion-minimal set of preference orderings  $\mathcal{Q}$  whose inclusion in  $\mathcal{P}$  makes the resulting profile  $\overline{\mathcal{P}} = \mathcal{P} \uplus \mathcal{Q}$  single crossing with respect to some tree. We call the profile  $\overline{\mathcal{P}}$  the *closure* of  $\mathcal{P}$ . Using this uniqueness of closure property of weakly single crossing preference profiles on trees and the known recognition algorithm of single crossing preference profiles on trees due to Clearwater et al. [CPS15], we design a polynomial time algorithm for recognizing weakly single crossing profiles on trees. We then move on to develop a polynomial time algorithm with low query complexity for eliciting weakly single crossing preference profiles on trees even when we do not know any tree with respect to which the closure of the input preference profile is single crossing and the preferences can be queried only sequentially; moreover, the sequential order of the input preference profile is also unknown. We complement the performance of our preference elicitation algorithm by proving that our algorithm makes an optimal number of queries up to constant factors when the number of preference orderings in the input preference profile is large compared to the number of candidates, even if the

input preference profile is known to be single crossing with respect to some given tree and the preference orderings can be accessed randomly.

# 1 Introduction

Aggregating preferences of a set of agents is a common problem in multiagent systems. In a typical setting, we have a set of  $m$  candidates, a set of  $n$  voters each of whom possesses a *preference* which is a linear order (reflexive, anti-symmetric, and transitive relations) over the set of candidates, and we would like to aggregate these preferences into one preference which intuitively “reflects” the preference of all the voters. However, it is well-known that the pairwise majority relation of a set of preferences may often be intransitive due to existence of *Condorcet cycles* – a set of candidates  $\{c_i : i \in [\ell]\}$  where  $c_i$  is preferred over  $c_{i \pmod{\ell} + 1}$  by a majority of the voters for every  $i \in [\ell]$  (see Moulin [Mou91]). Consecutively, a substantial amount of research effort has been devoted to finding interesting restrictions on the preferences of the voters which ensure transitivity of the majority relation (see Merlin and Gaertner [MG04] for a survey).

Among the most widely used domains in social choice theory are the *single peaked* and *single crossing* domains. Introduced by Black [Bla48], the single peaked domain not only satisfies transitivity of majority relation but also captures the essence of many election scenarios including political elections [HM97]. Intuitively, a *profile*, which is a tuple of preferences of all the voters, is called single peaked if the candidates can be arranged in a linear order (often called societal axis or harmonious order) and every voter can be placed somewhere in that linear order so that she prefers candidates “closer” to her than candidates “far” from her. The notion of single peakedness has subsequently been generalized further, often at the cost of the transitivity property of majority relation. For example, the popular single peaked domain on trees [Dem82] only guarantees existence of a *Condorcet winner* (for an odd number of voters) and the notion of single peaked width [CGS12] does not even guarantee existence of a Condorcet winner. A Condorcet winner is a candidate who is preferred over every other candidate by a majority of the voters.

Mirrlees [Mir71] proposed the single crossing domain where voters (instead of candidates as in the case of single peaked domain) can be arranged in a linear order so that, for every two candidates  $x$  and  $y$ , all the voters who prefer  $x$  over  $y$  appear consecutively. Other than guaranteeing transitivity property of the majority relation, the single crossing domain has found wide applications in economics [DS74], specially in redistributive income taxation [Rob77, MR81], trade union bargaining [BC84], etc. The notion of single crossingness has further been generalized with respect to *median graphs* while maintaining the transitivity of the majority relation [Dem12]. A graph is called a median graph if, for every three nodes, there exists a unique node which is present in the shortest paths between all three pairs of nodes. Trees, hypercubes, etc. are important examples of median graphs. In recent times, Kung applied single-crossingness on trees to the study of networks [Kun15].

All the domains discussed above also enjoy other desirable properties. Existence of strategy-proof voting rules, polynomial time tractability of the problem of finding the winner for important voting rules like Kemeny [Kem59, Lev75, BBHH15], Dodgson [Dod76, BNM<sup>+</sup>58], Chamberlin-Courant [BSU13, SYFE13, CPS15], etc. are prominent examples of such properties. Domain restrictions in the presence of incomplete votes (where votes may be partial orders instead of complete orders) has also received significant research attention in recent times. Lackner [Lac14] showed that determining whether a set of incomplete votes is single peaked is a NP-complete problem. Elkind et al. [EFLO15a] performed similar exercise for the single crossing domain and showed that although the problem of finding whether a set of incomplete votes is single crossing is NP-complete in general, it admits polynomial time algorithms for top orders and for various other practically appealing cases. Elkind and Lackner [EL15] extended the notions of single peakedness and single crossingness to approval votes and showed that various computational problems that are hard in the approval voting setting become polynomial time tractable. In a approval voting scenario, every vote simply approves a subset of candidates instead of specifying a complete ranking of the candidates. Closeness of a profile to various domains also forms an active body of current research. The work of Erdelyi et al. [ELP13] and Brederick et al. [BCW16] showed that the problem of finding the distance of a profile from the single peaked and single crossing domains are often NP-hard and occasionally polynomial time solvable under various natural notions of distance. Elkind and Lackner [EL14] provided approximation and fixed parameter tractable algorithms for the problems of finding the minimum number of votes/candidates that need to be deleted so that the resulting profile belongs to some specific domain – their algorithms work for any domain which can be characterized by forbidden configurations and thus, in particular, for both the single peaked and single crossing domains. Faliszewski et al. [FHH14] studied various manipulation, control, and bribery problems in nearly single peaked domain and showed many interesting results; for example, they proved that some control problems suddenly become NP-hard even in the presence of only one maverick whereas many other manipulation problems continue to be polynomial time solvable with a reasonable number of mavericks. We refer the reader to the recent survey article by Elkind et al. [ELP16] and references therein for a more complete picture of recent research activity in preference restrictions in computational social choice theory.

## 1.1 Motivation and Related Work

A property that is usually true for all popular domains is what we call “downward monotonicity” – if a profile belongs to a domain  $\mathcal{D}$ , then so is any subset of the profile (votes). To the best of our knowledge, the only important exception to this is the single crossing domain on trees [CPS14]. Interestingly, the sub-profiles of a single crossing profile on trees continue to exhibit desirable properties like transitivity of majority relation, polynomial time recognition algorithm [CPS15], existence of polynomial time algorithms for Kemeny and Dodgson voting rules, etc. This makes the study of the sub-profiles of single

crossing profiles on trees important. We call the domain consisting of all profiles which are sub-profiles of some single crossing profile on trees, *weakly single crossing domain on trees*. We remark that the domain of top monotonic profiles also does not satisfy similar notion of downward monotonicity with respect to candidates – deleting candidates from a top monotonic profile may destroy top monotonicity property. However, the votes are much more general there than simply being complete rankings and are allowed to be any complete, reflexive, and transitive (not necessarily anti-symmetric) binary relations. We refer the interested readers to the work of Barbera and Moreno [BM11] for an elaborate exposition to top monotonic profiles. We also remark that the idea of defining a domain as a set of profiles that are obtained by deleting some votes from another profile of a different domain has been used by Elkind et al. [EFS14] to characterize profiles that are simultaneously single peaked and single crossing.

One of the first questions to ask for any domain is whether there exists an efficient recognition algorithm – given a profile  $\mathcal{P}$ , does there exist a polynomial time algorithm to decide whether  $\mathcal{P}$  belongs to the domain? Indeed, there exist efficient recognition algorithms for many popular domains, for example, single peaked [Tri89], single crossing [BCW13, CPS15], and so on [Kno10, EFLO15b, EF14]. This motivates us to pursue a similar question for the weakly single crossing domain on trees – does there exist a polynomial time recognition algorithm for the weakly single crossing domain on trees?

Another fundamental problem in social choice theory is preference elicitation – elicit the preferences of a set of agents by asking them (hopefully a small number of) comparison queries. Indeed, in many applications of social choice theory, for example, metasearch engines [DKNS01], spam detection [CSS99], computational biology [JSA08], the number of candidates is huge and thus, simply asking the agents to reveal their preferences is impractical. Conitzer [Con09] showed that, for the domain of single peaked profiles, we can elicit the preferences of a set of agents by asking a small number of comparison queries. Recently, a similar study has been carried out for the single peaked domain on trees [DM16a] and single crossing domain [DM16b]. This motivates us to study preference elicitation for the weakly single crossing domain on trees.

## 1.2 Our Contribution

Our specific contributions in this work are as follows.

- ▷ Given a weakly single crossing profile  $\mathcal{P}$  on trees, we show that there exists an inclusion-minimal profile  $\overline{\mathcal{P}}$  which contains  $\mathcal{P}$  and is itself single crossing with respect to some tree [Theorem 1]. We call  $\overline{\mathcal{P}}$  the *single crossing tree closure* of  $\mathcal{P}$ .
- ▷ We extend the polynomial time recognition algorithm of Clearwater et al. [CPS15] for single crossing profiles on trees to a polynomial time recognition algorithm for weakly single crossing profiles on trees. Moreover, for any weakly single crossing profile  $\mathcal{P}$  on trees, our recognition algorithm also outputs a tree  $\mathcal{T}$  on the single crossing tree closure  $\overline{\mathcal{P}}$  of  $\mathcal{P}$  with respect to which  $\overline{\mathcal{P}}$  is single crossing [Corollary 1].

- ▷ We present a polynomial time algorithm with query complexity  $\mathcal{O}(mn + \min\{m^2, n\}m \log m)$  for eliciting weakly single crossing profiles on trees, even if we do not a priori know any tree with respect to which the single crossing tree closure of the input profile is single crossing and we are only allowed to access the preferences in an arbitrary (unknown) sequential order [Theorem 3 and Corollary 2]. We wish mention that the query complexity of our preference elicitation algorithm matches (up to constant factors) with the query complexity of preference elicitation algorithm for single crossing profiles due to Dey and Misra [DM16b].
- ▷ We complement the query complexity of our preference elicitation algorithm by showing that any preference elicitation algorithm for single crossing profiles on trees has query complexity  $\Omega(mn + m \log m)$ , even if the input profile is single crossing with respect to a known tree and we are allowed to query preferences randomly and interleave the queries to different preferences arbitrarily [Theorem 4].

## 2 Preliminaries

For a positive integer  $\ell$ , we denote the set  $\{1, \dots, \ell\}$  by  $[\ell]$ . For a set  $\mathcal{X}$  and an integer  $k$ , we denote the set of all possible subsets of  $\mathcal{X}$  of size  $k$  by  $\mathfrak{P}_k(\mathcal{X})$ .

Let  $\mathcal{V} = \{v_i : i \in [n]\}$  be a set of  $n$  voters and  $\mathcal{C} = \{c_j : j \in [m]\}$  be a set of  $m$  candidates. If not mentioned otherwise, we denote the set of candidates, the set of voters, the number of candidates, and the number of voters by  $\mathcal{C}$ ,  $\mathcal{V}$ ,  $m$ , and  $n$  respectively. Every voter  $v_i$  has a *preference*  $\succ_i$  which is a complete order over the set  $\mathcal{C}$  of candidates. We say voter  $v_i$  prefers a candidate  $x \in \mathcal{C}$  over another candidate  $y \in \mathcal{C}$  if  $x \succ_i y$ . We denote the set of all preferences over  $\mathcal{C}$  by  $\mathcal{L}(\mathcal{C})$ . The  $n$ -tuple  $(\succ_i)_{i \in [n]} \in \mathcal{L}(\mathcal{C})^n$  of the preferences of all the voters is called a *profile*. For a subset  $M \subseteq [n]$ , we call  $(\succ_i)_{i \in M}$  a sub-profile of  $(\succ_i)_{i \in [n]}$ . We often view a profile  $\mathcal{P}$  as a multi-set consisting of the preferences in  $\mathcal{P}$  to avoid use of cumbersome notations. The view of a profile we are considering will be clear from the context. Given a preference  $\succ \in \mathcal{L}(\mathcal{C})$  and a positive integer  $\ell$ , we denote the profile consisting of  $\ell$  copies of the preference  $\succ$  by  $\succ^\ell$ . A *domain* is a set of profiles. The single crossing domain is defined as follows.

**Definition 1** (Single Crossing Domain). A profile  $\mathcal{P} = (\succ_i)_{i \in [n]}$  of  $n$  preferences over a set  $\mathcal{C}$  of candidates is called a *single crossing profile* if there exists a permutation  $\sigma$  of  $[n]$  such that, for every two distinct candidates  $x, y \in \mathcal{C}$ , whenever we have  $x \succ_{\sigma(i)} y$  and  $x \succ_{\sigma(j)} y$  for two integers  $i$  and  $j$  with  $1 \leq \sigma(i) < \sigma(j) \leq n$ , we have  $x \succ_{\sigma(k)} y$  for every  $\sigma(i) \leq k \leq \sigma(j)$ .

Demange significantly generalizes the single crossing domain to the single crossing domain on median graphs in [Dem12]. A graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is called a *median graph* if, for every three nodes  $u, v, w \in \mathcal{V}$ , there exists a unique vertex  $m(u, v, w)$ , called the median of the nodes  $u, v$ , and  $w$ , which is present in the shortest paths between each pair of  $u, v$ , and  $w$ . Trees and hypercubes are important examples of median graphs. In this work, we will be concerned with trees only. A *tree* is a connected acyclic graph. A *star* is a tree where

there is a *central node* with whom every other node is connected by an edge. We refer to [Die97] for common terminologies of trees. Given a tree  $\mathcal{T}$  and a node  $u \in \mathcal{T}$ , we denote the tree rooted at  $u$  by  $\mathcal{T}[u]$ . The single crossing domain on trees is defined as follows.

**Definition 2** (Single Crossing Domain on Trees). *Let  $\mathcal{T}$  be a tree with the set of voters as the set of nodes of  $\mathcal{T}$ . A profile  $\mathcal{P}$  is called single crossing with respect to a tree  $\mathcal{T}$  if every sub-profile along every path of  $\mathcal{T}$  is single crossing.*

Let a profile  $\mathcal{P}$  be single crossing with respect to a tree  $\mathcal{T}$ . We call  $\mathcal{T}$  a *single crossing tree* of  $\mathcal{P}$ . We denote the preference associated with a voter (which is a node in  $\mathcal{T}$ )  $u \in \mathcal{T}$  by  $\succ_u$ . We denote a voter in  $\mathcal{T}$  associated with a preference  $\succ \in \mathcal{P}$  by  $u_\succ$ . An equivalent condition for a profile to be single crossing with respect to a tree  $\mathcal{T}$  is that, for every pair of candidates  $x, y \in \mathcal{C}$ , there exists at most one edge in the cut  $(\mathcal{V}_{x \succ y}, \mathcal{V} \setminus \mathcal{V}_{x \succ y})$ , where  $\mathcal{V}_{x \succ y}$  is the set of voters who prefer  $x$  over  $y$  [CPS15]. We now generalize the single crossing domain on trees to the weakly single crossing domain on trees as follows.

**Definition 3** (Weakly Single Crossing Domain on Trees). *The weakly single crossing domain on trees is the set of all profiles  $\mathcal{P}$  such that there exists a profile  $\mathcal{P}'$  which contains  $\mathcal{P}$  and is itself single crossing with respect to a tree  $\mathcal{T}$ .*

## 2.1 Problem Formulation

We study the following problem for recognizing weakly single crossing profiles on trees.

**Problem Definition 1** (WEAKLY SINGLE CROSSING TREE RECOGNITION). *Given a profile  $\mathcal{P}$ , does  $\mathcal{P}$  belong to the weakly single crossing domain on trees?*

Suppose we have a profile  $\mathcal{P}$  with  $n$  voters and  $m$  candidates. Let us define a function  $\text{QUERY}(x \succ_\ell y)$  for a voter  $\ell$  and two different candidates  $x$  and  $y$  to be TRUE if the voter  $\ell$  prefers the candidate  $x$  over the candidate  $y$  and FALSE otherwise. We now define the preference elicitation problem.

**Problem Definition 2** (PREFERENCE ELICITATION). *Given an oracle access to  $\text{QUERY}(\cdot)$  for a profile  $\mathcal{P}$ , find  $\mathcal{P}$ .*

For two distinct candidates  $x, y \in \mathcal{C}$  and a voter  $\ell$ , we say a PREFERENCE ELICITATION algorithm  $\mathcal{A}$  compares candidates  $x$  and  $y$  for the voter  $\ell$ , if  $\mathcal{A}$  makes a call to either  $\text{QUERY}(x \succ_\ell y)$  or  $\text{QUERY}(y \succ_\ell x)$ . We define the number of queries made by the algorithm  $\mathcal{A}$ , called the *query complexity* of  $\mathcal{A}$ , to be the number of distinct tuples  $(\ell, x, y) \in \mathcal{V} \times \mathcal{C} \times \mathcal{C}$  with  $x \neq y$  such that the algorithm  $\mathcal{A}$  compares the candidates  $x$  and  $y$  for the voter  $\ell$ . Notice that, even if the algorithm  $\mathcal{A}$  makes multiple calls to  $\text{QUERY}(\cdot)$  with same tuple  $(\ell, x, y)$ , we count it only once in the query complexity of  $\mathcal{A}$ . This is without loss of generality since we can always implement a “wrapper” around the oracle which memorizes all the calls made to the oracle so far and whenever it receives a duplicate call, it replies from its memory without “actually” making a call to the oracle.

The following observation is immediate from standard sorting algorithms like merge sort.



**Observation 1.** *There is a PREFERENCE ELICITATION algorithm for eliciting one preference with query complexity  $\mathcal{O}(m \log m)$ .*

**Model of Input for PREFERENCE ELICITATION** There are two prominent models for accessing the preferences in the literature (see [DM16b]). In the *random access model*, we are allowed to query any preference at any point of time. Moreover, we are also allowed to interleave the queries to different voters. In the *sequential access model*, voters are arriving in a sequential manner one after another to the system. Once a voter  $\ell$  arrives, we can query her preference as many times as we like and then we “release” the voter  $\ell$  from the system to grab the next voter in the queue. Once the voter  $\ell$  is released, its preference can never be queried again.

### 3 Recognizing Weakly Single Crossing Profiles on Trees

In this section, we present our polynomial time algorithm for the WEAKLY SINGLE CROSSING TREE RECOGNITION problem. We begin with bounding the number of distinct preferences in a profile which is single crossing with respect to some tree. We remark that the exact same bound of Lemma 1 is known for the single crossing profiles [DM16b].

**Lemma 1.** *Let a profile  $\mathcal{P}$  of distinct preferences be single crossing with respect to a tree  $\mathcal{T}$ . Then  $|\mathcal{P}| \leq \binom{m}{2} + 1$ .*

*Proof.* Let  $e = (u, v) \in \mathcal{T}$  be any edge of  $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ . Since the preferences in  $\mathcal{P}$  are all distinct, the set  $\mathcal{D}_e = \{\{x, y\} \in \mathfrak{P}_2(\mathcal{C}) : \succ_u \text{ and } \succ_v \text{ order } x \text{ and } y \text{ differently}\}$  of pairs of candidates that are ordered differently by the voters  $u$  and  $v$  is nonempty. Since the profile  $\mathcal{P}$  is single crossing with respect to  $\mathcal{T}$ , for any two distinct edges  $e_1, e_2 \in \mathcal{T}$ , we have  $\mathcal{D}_{e_1} \cap \mathcal{D}_{e_2} = \emptyset$ . We also have  $\cup_{e \in \mathcal{T}} \mathcal{D}_e \subseteq \mathfrak{P}_2(\mathcal{C})$ . Now we bound  $|\mathcal{P}|$  as follows.

$$|\mathcal{P}| = |\mathcal{E}| + 1 \leq \binom{m}{2} + 1$$

□

The following result which is immediate from the proof of Lemma 3.7 in [CPS15], simplifies lot of our proofs.

**Lemma 2.** *Let  $\mathcal{P} = (\succ_i^{\ell_i})_{i \in [n]}$ ,  $\ell_i > 0, \forall i \in [n]$ , be a profile with  $\succ_i \neq \succ_j, \forall i \neq j$  and  $\mathcal{P}' = (\succ_i)_{i \in [n]}$  be the profile resulting from  $\mathcal{P}$  after removing all the duplicate preferences. Then  $\mathcal{P}$  is single crossing with respect to some tree if and only if  $\mathcal{P}'$  is single crossing with respect to some (other) tree. Therefore,  $\mathcal{P}$  is a weakly single crossing profile on trees if and only if  $\mathcal{P}'$  is a weakly single crossing profile on trees. Moreover, given the tree with respect to which the profile  $\mathcal{P}$  is single crossing, we can construct another tree with respect to which  $\mathcal{P}'$  is single crossing in polynomial amount of time and vice versa.*



We next define the majority relation of a set of preferences which helps us formulate an important property of single crossing domain with respect to trees. We call that property the *triad majority property*. This property will in turn help us in defining and finding what we call the *single crossing tree closure* of a weakly single crossing profile on trees. The notion of a single crossing tree closure plays a central role in our recognition and elicitation algorithms.

**Definition 4** (Majority Relation). *Given a profile of  $n$  preferences  $(\succ_i)_{i \in [n]} \in \mathcal{L}(\mathcal{C})^n$ , we call the relation  $\succ = \text{Maj}((\succ_i)_{i \in [n]}) = \{x \succ y : x, y \in \mathcal{C}, |\{i \in [n] : x \succ_i y\}| > n/2\}$  the majority relation of the profile  $(\succ_i)_{i \in [n]}$ . If the majority relation  $\text{Maj}(\mathcal{P})$  of a profile  $\mathcal{P}$  turns out to be a linear order, we say that a majority order of  $\mathcal{P}$  exists and we call  $\text{Maj}(\mathcal{P})$  the majority order of  $\mathcal{P}$ .*

**Property 1** (Triad Majority Property). *A profile  $\mathcal{P}$  is said to satisfy the triad majority property if, for every three preferences (not necessarily distinct)  $\succ_i \in \mathcal{P}, i \in [3]$ , the majority relation  $\text{Maj}((\succ_i)_{i \in [3]})$  of  $(\succ_i)_{i \in [3]}$  is a linear order and  $\text{Maj}((\succ_i)_{i \in [3]}) \in \mathcal{P}$ .*

We now show that every single crossing profile on trees, satisfies the triad majority property.

**Lemma 3.** *Let  $\mathcal{P}$  be a single crossing profile with respect to a tree  $\mathcal{T}$ . Then  $\mathcal{P}$  satisfies the triad majority property.*

*Proof.* For any three preferences  $\succ_i, i \in [3]$ , let the nodes associated with  $\succ_i \in \mathcal{P}, i \in [3]$ , in  $\mathcal{T}$  be  $u_{\succ_i} \in \mathcal{T}, i \in [3]$ . Let  $u$  be the unique node in  $\mathcal{T}$  that lies in the shortest paths  $u_{\succ_1}$  to  $u_{\succ_2}$ ,  $u_{\succ_2}$  to  $u_{\succ_3}$ , and  $u_{\succ_3}$  to  $u_{\succ_1}$  in  $\mathcal{T}$ . We claim that  $\text{Maj}(\succ_1, \succ_2, \succ_3) = \succ_u$ , where  $\succ_u$  is the preference associated with the node  $u$ . Suppose not, then there exist two candidates  $x, y \in \mathcal{C}$  such that  $x \succ_u y, y \succ_i x$ , and  $y \succ_j x$  for  $i, j \in [3], i \neq j$ . But then the sub-profile of  $\mathcal{P}$  along the path between  $u_{\succ_i}, u_{\succ_j}$  is not single crossing. This contradicts our assumption that  $\mathcal{P}$  is single crossing with respect to the tree  $\mathcal{T}$ .  $\square$

We now define the notion of *triad majority closure* of a profile which will be used crucially in our algorithm for WEAKLY SINGLE CROSSING TREE RECOGNITION.

**Definition 5** (Triad Majority Closure). *Let  $\mathcal{P}'$  be a profile that satisfies the triad majority property and  $\mathcal{P}$  be a sub-profile of  $\mathcal{P}'$ . Then the triad majority closure of  $\mathcal{P}$  is defined to be the inclusion-wise minimal profile  $\overline{\mathcal{P}}$  that satisfies the triad majority property and contains  $\mathcal{P}$ .*

The following result shows that a triad majority closure of a sub-profile of a profile that satisfies the triad majority property, is unique thereby establishing the well-definedness of Definition 5.

**Proposition 1.** *Let  $\mathcal{P}'$  be a profile that satisfies the triad majority property and  $\mathcal{P}$  be a sub-profile of  $\mathcal{P}'$ . Let  $\overline{\mathcal{P}}$  be the triad majority closure of  $\mathcal{P}$  and  $\mathcal{M}(\mathcal{P})$  be the set of profiles which contain  $\mathcal{P}$  and satisfy the triad majority property. Then  $\overline{\mathcal{P}} = \cap_{\mathcal{P}' \in \mathcal{M}(\mathcal{P})} \mathcal{P}'$ . Hence, the triad majority closure of  $\mathcal{P}$  exists and it is unique.*

*Proof.* Follows from the fact that, for two profiles  $\mathcal{P}_1, \mathcal{P}_2$  that satisfy the triad majority property and contain  $\mathcal{P}$ , the profile  $\mathcal{P}_1 \cap \mathcal{P}_2$  also satisfies the triad majority property and contains  $\mathcal{P}$ .  $\square$

Similar to the triad majority closure, we next define the *single crossing tree closure*.

**Definition 6** (Single Crossing Tree Closure). *Let  $\mathcal{P}$  be a weakly single crossing profile on trees. The single crossing tree closure of  $\mathcal{P}$  is defined to be the inclusion-wise minimal profile  $\overline{\mathcal{P}}$  which is single crossing with respect to some tree.*

The following result shows that, for every weakly single crossing profile  $\mathcal{P}$  on trees, the single crossing tree closure  $\overline{\mathcal{P}}$  of  $\mathcal{P}$  exists and is unique, thereby establishing well-definedness of Definition 6. Moreover,  $\overline{\mathcal{P}}$  turns out to be the triad majority closure of  $\mathcal{P}$ .

**Theorem 1.** *Let  $\mathcal{P}$  be a weakly single crossing profile on trees. Then the triad majority closure  $\overline{\mathcal{P}}$  of  $\mathcal{P}$  is also single crossing with respect to some tree. Therefore, the single crossing tree closure of  $\mathcal{P}$  exists, it is unique, and equals to its triad majority closure  $\overline{\mathcal{P}}$ . Moreover, the single crossing tree closure of  $\mathcal{P}$  can be computed in polynomial time.*

*Proof.* We can assume, without loss of generality, that the preferences in  $\mathcal{P}$  are all distinct due to Lemma 2. Let  $\mathcal{R}$  be a profile of distinct preferences that is single crossing with respect to a tree  $\mathcal{T}$  and  $\mathcal{P}$  be a sub-profile of  $\mathcal{R}$  – such an  $\mathcal{R}$  exists since  $\mathcal{P}$  is a weakly single crossing profile on trees. Let us consider the subgraph  $\mathcal{T}' = \bigcup_{\succ, \succ' \in \mathcal{P}} \mathbf{p}_{u_{\succ}, u_{\succ'}}$  of  $\mathcal{T}$ , where, for two nodes  $u, v \in \mathcal{T}$ ,  $\mathbf{p}_{u,v}$  denotes the unique path between  $u$  and  $v$  in  $\mathcal{T}$ . We observe that  $\mathcal{T}'$  is actually a subtree of  $\mathcal{T}$  since it is a connected subgraph of the tree  $\mathcal{T}$ . We also observe that, for every leaf node  $l$  of  $\mathcal{T}'$ , the preference  $\succ_l$  corresponding to  $l$  always belongs to  $\mathcal{P}$ . We iteratively apply the following transformation on  $\mathcal{T}'$  as long as we can: if there exists a node  $u$  in  $\mathcal{T}'$  of degree two such that the preference  $\succ_u$  associated with  $u$  does not belong to  $\mathcal{P}$ , then we “bypass”  $u$ , that is, we remove  $u$  from  $\mathcal{T}'$  along with the two edges incident on  $u$  and add an edge between the two neighbors of  $u$ . Let us call the tree  $\mathcal{T}''$  which results from  $\mathcal{T}'$  after making all the transformations iteratively as long as we can. We claim that the triad majority closure  $\overline{\mathcal{P}}$  of  $\mathcal{P}$  is exactly the set  $\mathcal{Q}$  of preferences associated with the nodes of the tree  $\mathcal{T}''$ . We first observe that  $\mathcal{Q}$  satisfies the triad majority property since  $\mathcal{Q}$  is single crossing with respect to the tree  $\mathcal{T}''$  [see Lemma 3]. Now to show that  $\mathcal{Q}$  is the triad majority closure of  $\mathcal{P}$ , it is enough to show that every preference  $\succ \in \mathcal{Q} \setminus \mathcal{P}$  is the majority order of some three preferences in  $\mathcal{P}$ . Let  $\succ \in \mathcal{Q} \setminus \mathcal{P}$  and  $u \in \mathcal{T}''$  be the node in  $\mathcal{T}''$  whose corresponding preference is  $\succ$ . We observe that the degree of  $u$  in  $\mathcal{T}''$  is at least 3, since otherwise the transformation would have deleted  $u$ . So the degree of  $u$  in  $\mathcal{T}''$  is at least 3. We make  $\mathcal{T}''$  rooted at  $u$  and call it  $\mathcal{T}''[u]$ . Let  $u_i, i \in [3]$ , be any three children of  $u$  in  $\mathcal{T}''$ . Let the subtrees of the rooted tree  $\mathcal{T}''[u]$  rooted at  $u_i, i \in [3]$ , be  $\mathcal{T}_i, i \in [3]$ . We observe that there must exist a node  $v_i$  in  $\mathcal{T}_i$  such that the preference  $\succ_i$  attached to  $v_i$  belongs to  $\mathcal{P}$  for every  $i \in [3]$ . We claim that  $\succ$  is the majority order of  $(\succ_i)_{i \in [3]}$ . Indeed, otherwise we may assume (by renaming) that there exists two candidates  $x, y \in \mathcal{C}$  such that  $x \succ_1 y, x \succ_u y, y \succ_2 x, y \succ_3 x$ . However, this violates single crossing property on the

path between  $v_2$  and  $v_3$ . Hence  $\succ$  is the majority order of  $(\succ_i)_{i \in [3]}$ . This proves the claim. Hence,  $\mathcal{Q}$  is the triad majority closure of  $\mathcal{P}$ . Also  $\mathcal{Q}$  is single crossing with respect to the tree  $\mathcal{T}''$ . Hence, by Lemma 3,  $\mathcal{Q}$  is the single crossing tree closure of  $\mathcal{P}$ .

We consider the algorithm  $\mathcal{A}$  which iteratively adds  $\text{Maj}(\succ, \succ', \succ'')$  to  $\mathcal{Q}$  (which is initialized to empty set) for all three (not necessarily distinct) preferences  $\succ, \succ', \succ'' \in \mathcal{P}$  and outputs  $\mathcal{Q}$ . The algorithm  $\mathcal{A}$  runs in polynomial amount of time. The correctness of  $\mathcal{A}$  follows from the first part of the result.  $\square$

The following result on recognizing single crossing profiles on trees is due to [CPS15].

**Theorem 2.** (Theorem 4.2 in [CPS15]) *Given a profile  $\mathcal{P}$ , there is a polynomial time algorithm for checking whether there exists a tree  $\mathcal{T}$  with respect to which  $\mathcal{P}$  is single crossing. Moreover, if a single crossing tree exists, then the algorithm also outputs a tree  $\mathcal{T}'$  with respect to which  $\mathcal{P}$  is single crossing.*

Theorem 1 and 2 give us a polynomial time algorithm for the WEAKLY SINGLE CROSSING TREE RECOGNITION problem.

**Corollary 1.** *The WEAKLY SINGLE CROSSING TREE RECOGNITION problem is in P.*

*Proof.* We can assume, without loss of generality, that the preferences in  $\mathcal{P}$  are all distinct due to Lemma 2. Our algorithm  $\mathcal{A}$  first constructs the triad majority closure  $\overline{\mathcal{P}}$  of  $\mathcal{P}$  using the algorithm in Theorem 1. Our algorithm now checks whether  $\overline{\mathcal{P}}$  is single crossing with respect to some tree using Theorem 2. If the algorithm in Theorem 2 outputs YES, then  $\mathcal{A}$  also outputs YES and the tree returned by the algorithm in Theorem 2; otherwise  $\mathcal{A}$  outputs NO. The correctness of the algorithm follows from Theorem 1 and the correctness of the algorithm in Theorem 2. The polynomial running time of  $\mathcal{A}$  follows from polynomial running time of the algorithms in Theorem 1 and 2.  $\square$

## 4 Preference Elicitation for Weakly Single Crossing Profiles on Trees

In this section, we present a polynomial time low query complexity algorithm for eliciting weakly single crossing profiles on trees. We begin with bounding the degree of any node in a single crossing tree for a profile consisting of distinct preferences. We use this to bound the query complexity of our elicitation algorithm.

**Lemma 4.** *Let a profile  $\mathcal{P}$  of distinct preferences be single crossing with respect to a tree  $\mathcal{T}$ . Then the degree of every node in  $\mathcal{T}$  is at most  $m - 1$ .*

*Proof.* Consider any node  $v$  in  $\mathcal{T}$  and let the preference  $\succ_v$  associated with the node  $v$  be  $c_1 \succ c_2 \succ \dots \succ c_m$ . Let the neighbors of the node  $v$  in  $\mathcal{T}$  be  $u_i, i \in [\ell]$ , and the preferences associated with them be  $\succ_{u_i}, i \in [\ell]$ , respectively. We need to show that  $\ell \leq m - 1$ . We first observe that, for every  $i \in [m - 1]$ , there exists at most one child  $u_j, j \in [\ell]$ , of the

node  $v$  such that the preference  $\succ_{u_j}$  associated with the node  $u_j$  has  $c_{i+1} \succ_{u_j} c_i$ . Indeed, otherwise let us assume that there exist two children  $u_j$  and  $u_{j'}, j, j' \in [\ell], j \neq j'$ , of the node  $v$  such that the preferences  $\succ_{u_j}$  and  $\succ_{u_{j'}}$  associated with them both have  $c_{i+1} \succ_{u_j} c_i$  and  $c_{i+1} \succ_{u_{j'}} c_i$ . Then the sub-profile along the path  $u_j, v, u_{j'}$  is not single crossing which contradicts our assumption that  $\mathcal{P}$  is single crossing with respect to the tree  $\mathcal{T}$ . Hence, for every  $i \in [m-1]$ , there exists at most one child  $u_j, j \in [\ell]$ , of the node  $v$  such that the preference  $\succ_{u_j}$  associated with the node  $u_j$  has  $c_{i+1} \succ_{u_j} c_i$ . We also observe that, for every  $j \in [\ell]$ , there exists an  $i \in [m-1]$  such that  $c_{i+1} \succ_{u_j} c_i$  since otherwise we have  $\succ_v = \succ_{u_j}$  which contradicts our assumption that the preferences of the profile  $\mathcal{P}$  are all distinct. Hence, we have  $\ell \leq m-1$  since otherwise we will have  $i \in [m-1]$  and  $j, j' \in [\ell], j \neq j'$ , such that we have both  $c_{i+1} \succ_{u_j} c_i$  and  $c_{i+1} \succ_{u_{j'}} c_i$  due to pigeon hole argument which leads to a contradiction as argued above.  $\square$

We remark that the assumption that the profile in Lemma 4 consists of distinct preferences is crucial since a profile where all the preferences are the same is single crossing with respect to every tree. We show next a structural result about trees. We will use it crucially for bounding query complexity of our elicitation algorithm.

**Lemma 5.** *Let  $\mathcal{T}$  be a tree of size at least 3. Then there exists a node  $r$  in  $\mathcal{T}$  of degree at least 2 such that the following holds: make the tree  $\mathcal{T}$  rooted at  $r$ ; let the neighbors of  $r$  be  $u_1, \dots, u_\ell$ ; let the rooted subtrees of  $\mathcal{T}[r]$  rooted at  $u_i, i \in [\ell]$ , be respectively  $\mathcal{T}_i, i \in [\ell]$ , and  $|\mathcal{T}_1| \geq |\mathcal{T}_2| \geq \dots \geq |\mathcal{T}_\ell|$ ; then  $|\mathcal{T}_1| \leq 3|\mathcal{T}|/4$ .*

*Proof.* Let  $u_1$  be any arbitrary node in  $\mathcal{T}$ . If  $u_1$  satisfies the properties of the lemma, then we are done. Otherwise, let  $u_2$  be the neighbor of  $u_1$  such that, if we remove the edge  $(u_1, u_2)$ , then the connected component containing  $u_2$  has more than  $3|\mathcal{T}|/4$  nodes. Let the removal of the edge  $(u_1, u_2)$  creates two connected component  $\mathcal{V}_1$  and  $\mathcal{W}_1$  such that  $u_1 \in \mathcal{V}_1$ . If  $u_2$  satisfies the properties of the lemma, then we are done. Otherwise, we repeat the above process defining  $u_3$  and  $\mathcal{V}_2$ . We observe that  $\mathcal{V}_1 \subsetneq \mathcal{V}_2$ . We continue this process until we get a node that satisfies the properties of the lemma. The process has to terminate because otherwise we have an infinite chain  $\mathcal{V}_1 \subsetneq \mathcal{V}_2 \subsetneq \mathcal{V}_3 \subsetneq \dots \subseteq \mathcal{T}$ . This cannot happen since all the inclusions in the chain above are proper however  $\mathcal{V}_i \subseteq \mathcal{T}$  for every  $i$ . Hence, the process always terminates with a node satisfying the properties of the lemma.  $\square$

Given a profile  $\mathcal{P}$  of distinct preferences, a tree  $\mathcal{T}$  with respect to which  $\mathcal{P}$  is single crossing, and a preference  $\succ \in \mathcal{L}(\mathcal{C})$ , we now present an algorithm for finding whether  $\succ$  belongs to  $\mathcal{T}$  using  $\mathcal{O}(m)$  calls to  $\text{QUERY}(\cdot)$ . This algorithm is a crucial component in our PREFERENCE ELICITATION algorithm.

**Lemma 6.** *Given a profile  $\mathcal{P}$  of distinct preferences, a tree  $\mathcal{T}$  with respect to which  $\mathcal{P}$  is single crossing, and an oracle access to a preference  $\succ \in \mathcal{L}(\mathcal{C})$ , there exists a polynomial time algorithm for finding whether  $\succ$  belongs to  $\mathcal{T}$  using  $\mathcal{O}(m)$  calls to  $\text{QUERY}(\cdot)$ .*

---

**Algorithm 1** for searching preference in a single crossing tree

---

**Input:** A profile  $\mathcal{P} = (\succ_i)_{i \in [n]}$  of distinct preferences, a tree  $\mathcal{T}$  with respect to which  $\mathcal{P}$  is single crossing, and a preference  $\succ \in \mathcal{L}(\mathcal{C})$

**Output:** YES if  $\succ$  belongs to  $\mathcal{P}$  and NO otherwise

```
1: while  $|\mathcal{T}| \geq 3$  do
2:    $r \leftarrow$  a node in  $\mathcal{T}$  as in Lemma 5.  $u_i, i \in [\ell], \ell \geq 2$ , be the neighbors of  $r$ . Let
    $\mathcal{T}_i, i \in [\ell]$ , be the subtrees of  $\mathcal{T}[r]$  rooted at  $u_i, i \in [\ell]$ , respectively such that  $|\mathcal{T}_1| \geq$ 
    $|\mathcal{T}_2| \geq \dots \geq |\mathcal{T}_\ell|$ .
3:   for  $i = 1$  to  $\ell$  do
4:     Let  $x_i, y_i \in \mathcal{C}$  such that  $x_i \succ_r y_i$  and  $y_i \succ_{u_i} x_i$ 
5:     if QUERY  $(x_i \succ y_i) = \text{TRUE}$  then
6:        $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_i$ 
7:     else
8:        $\mathcal{T} \leftarrow \mathcal{T}_i$  and exit for loop
9:     end if
10:  end for
11: end while
12: if  $\succ$  belongs to  $\mathcal{T}$  then ▷ Can be done in  $\mathcal{O}(m)$  queries.
13:   return YES
14: else
15:   return NO
16: end if
```

---

*Proof.* We present our algorithm in Algorithm 1. Let  $r$  be a node as in Lemma 5. We consider the tree  $\mathcal{T}$  rooted at  $r$  which we denote by  $\mathcal{T}[r]$ . Let the children of  $r$  in  $\mathcal{T}[r]$  be  $u_1, \dots, u_\ell$ , for some  $2 \leq \ell \leq m - 1$  (the upper bound of  $m - 1$  follows from Lemma 4), the rooted subtrees of  $\mathcal{T}[r]$  rooted at  $u_i, i \in [\ell]$ , be  $\mathcal{T}_i, i \in [\ell], |\mathcal{T}_1| \geq |\mathcal{T}_2| \geq \dots \geq |\mathcal{T}_\ell|$ , and  $|\mathcal{T}_1| \leq 3|\mathcal{T}|/4$ . Since the profile  $\mathcal{P}$  consists of distinct preferences, for every  $i \in [\ell]$ , there exist two candidates  $x_i, y_i \in \mathcal{C}$  such that  $x_i \succ_r y_i$  and  $y_i \succ_{u_i} x_i$ . Let  $\succ$  be the preference which we have to search in  $\mathcal{T}$ ; we can access  $\succ$  only through the QUERY  $(\cdot)$  function. We query the oracle for  $x_i$  versus  $y_i$  in  $\succ$ . If  $x_i \succ y_i$ , then  $\succ$  cannot belong to the subtree  $\mathcal{T}_i$  and thus we remove  $\mathcal{T}_i$  from  $\mathcal{T}$  as done in line 6 of Algorithm 1. On the other hand, if  $y_i \succ x_i$ , then  $\succ$  cannot belong to  $\mathcal{T} \setminus \mathcal{T}_i$  and thus we remove  $\mathcal{T} \setminus \mathcal{T}_i$  from  $\mathcal{T}$  as is done in line 8 of Algorithm 1. Hence, after each iteration of the while loop,  $\succ$  can only belong to  $\mathcal{T}$  and every iteration decreases the size of  $\mathcal{T}$ ; moreover,  $\succ$  can only belong to  $\mathcal{T}$ . Hence, the algorithm terminates and is correct. We now turn our attention to the query complexity of Algorithm 1. For a tree  $\mathcal{T}$  with  $n$  nodes where every preference is over  $m$  candidates, let  $T(n, m)$  be the query complexity of the while loop from line 1 to line 11 in Algorithm 1. Let us consider an iteration of the while loop at line 1. Let  $k$  be the number of times the for loop at line 3 iterates. If  $k = 1$ , then the algorithm makes only one query in the current iteration of the while loop in Algorithm 1 and the number of nodes in the new tree  $\mathcal{T}$  is upper bounded by 3/4-th time the number of nodes in the previous  $\mathcal{T}$  (by the choice of  $r$ ).

---

**Algorithm 2** for eliciting a profile which is single crossing with respect to some unknown tree

---

**Input:**  $\pi$  be the order in which voters arrive

**Output:** Profile of all the voters

```

1:  $\mathcal{R}, \mathcal{Q} \leftarrow \emptyset$ 
2:  $\triangleright \mathcal{Q}$  stores all the votes seen so far without duplicate.  $\mathcal{R}$  stores the profile.
3: for  $i \leftarrow 1$  to  $n$  do
4:    $\triangleright$  Elicit preference of the  $i^{th}$  voter in  $i^{th}$  iteration of this for loop.
5:    $\mathcal{T}$  be the tree with respect to which the single crossing tree closure  $\overline{\mathcal{Q}}$  of  $\mathcal{Q}$  is single crossing
6:    $\triangleright$  Using Theorem 1
7:   if  $\succ_{\pi(i)} = \succ$  for some  $\succ \in \overline{\mathcal{Q}}$  then
8:      $\triangleright$  Can be done using  $\mathcal{O}(m)$  queries by Lemma 6
9:      $\mathcal{R}[\pi(i)] \leftarrow \succ$ 
10:  else
11:     $\mathcal{R}[\pi(i)] \leftarrow$  Elicit using Observation 1
12:     $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathcal{R}[\pi(i)]\}$ 
13:  end if
14: end for

```

---

For  $k \geq 2$ , we observe that the number of nodes in the new tree  $\mathcal{T}$  is upper bounded by  $1/k$ -th times the number of nodes in the previous  $\mathcal{T}$ – this is because the for loop at line 3 iterates according to a nonincreasing order of subtrees. Hence, we have the following recurrence relation.

$$T(n, m) \leq \max\{T(3n/4, m) + 1, \max_{k=\{2, \dots, m\}} \{T(n/k) + k\}\}$$

$$T(2, m) = T(1, m) = \mathcal{O}(m)$$

We know from Lemma 1 that  $n \leq \binom{m}{2} + 1$ . By solving the above recurrence with the  $\binom{m}{2} + 1$  upper bound on  $n$  and the fact that line 12 to 16 can be executed with  $\mathcal{O}(m)$  queries (see [DM16b]), we get that the query complexity of Algorithm 1 is  $\mathcal{O}(m)$ .  $\square$

We first present a PREFERENCE ELICITATION algorithm for single crossing profiles on trees when we are given a sequential access to the preference (the sequential order is a priori not known) and we do not know any tree with respect to which the input profile is single crossing.

**Theorem 3.** *Suppose a profile  $\mathcal{P}$  is single crossing with respect to some unknown tree  $\mathcal{T}$ . Suppose we have only sequential access to the preferences whose ordering is also not known a priori. Then there is a PREFERENCE ELICITATION algorithm for the single crossing profiles on trees with query complexity  $\mathcal{O}(mn + \min\{m^2, n\}m \log m)$ .*



*Proof.* We present our PREFERENCE ELICITATION algorithm in Algorithm 2. We maintain an array  $\mathcal{R}$  of length  $n$  to store the preferences of all the  $n$  voters and a set  $\mathcal{Q}$  to store all the distinct preferences seen so far. Let  $\pi$  be the order in which the voters are accessed. To elicit the preference of voter  $\pi(i)$ , we first construct, in polynomial time, a tree  $\mathcal{T}$  with respect to which the single crossing tree closure  $\overline{\mathcal{Q}}$  of  $\mathcal{Q}$  is single crossing using Theorem 1. Next we find whether the preference of the voter  $\pi(i)$  already present in  $\mathcal{T}$ ; this can be done in polynomial time with making  $\mathcal{O}(m)$  calls to  $\text{QUERY}(\cdot)$  using Lemma 6. If the preference  $\succ_{\pi(i)}$  of the voter  $\pi(i)$  is present in  $\mathcal{T}$ , we have elicited  $\succ_{\pi(i)}$  using  $\mathcal{O}(m)$  queries. Otherwise, we elicit  $\succ_{\pi(i)}$  using  $\mathcal{O}(m \log m)$  queries by Observation 1. However, since the number of distinct preferences in any profile which is single crossing with respect to some tree is at most  $\binom{m}{2} + 1$  due to Lemma 1, we use Observation 1 at most  $\binom{m}{2} + 1$  times. Hence, the query complexity of Algorithm 2 is  $\mathcal{O}(mn + \min\{m^2, n\}m \log m)$ .  $\square$

Our algorithm in Theorem 3 can readily be seen to work for weakly single crossing profiles on trees too. Hence, we have the following corollary.

**Corollary 2.** *There exists a polynomial time PREFERENCE ELICITATION algorithm for the weakly single crossing profiles on trees with query complexity  $\mathcal{O}(mn + \min\{m^2, n\}m \log m)$ .*

We prove next that the query complexity upper bound in Theorem 3 is optimal up to constant factors for a large number of voters (more specifically, when  $n = \Omega(m^2 \log m)$ ), even if a tree  $\mathcal{T}$  is known with respect to which the input profile  $\mathcal{P}$  is single crossing and random access to preferences are allowed.

**Theorem 4.** *Let a profile  $\mathcal{P}$  be single crossing with respect to a tree  $\mathcal{T}$ . Let  $\mathcal{T}$  be known except for the preferences associated with the nodes of  $\mathcal{T}$ . Then any PREFERENCE ELICITATION algorithm has query complexity  $\Omega(m \log m + mn)$  even if we are allowed to query preferences randomly and interleave the queries to different preferences arbitrarily.*

*Proof.* The  $\Omega(m \log m)$  bound follows from the sorting lower bound and the fact that any profile consisting of only one preference  $P \in \mathcal{L}(\mathcal{C})$  is single crossing. Let  $\mathcal{T}$  be a star with one central node and  $n - 1$  leaf nodes attached to the central node with an edge. Suppose we have an even number of candidates, that is,  $\mathcal{C} = \{c_1, \dots, c_m\}$  for some even integer  $m$ . Consider the ordering  $\mathcal{Q} = c_1 \succ c_2 \succ \dots \succ c_m$  and the pairing of the candidates  $\{c_1, c_2\}, \{c_3, c_4\}, \dots, \{c_{m-1}, c_m\}$ . The oracle answers all the query requests consistently according to the ordering  $\mathcal{Q}$ . We claim that any PREFERENCE ELICITATION algorithm  $\mathcal{A}$  must compare  $c_i$  and  $c_{i+1}$  for every voter corresponding to every leaf node of  $\mathcal{T}$  and for every odd integer  $i \in [m]$ . Indeed, otherwise, there exist a voter  $\kappa$  corresponding to a leaf node of  $\mathcal{T}$  and an odd integer  $i \in [m]$  such that the algorithm  $\mathcal{A}$  does not compare  $c_i$  and  $c_{i+1}$ . Suppose the algorithm outputs a profile  $\mathcal{P}'$ . The oracle fixes the preference of every voter except  $\kappa$  to be  $\mathcal{Q}$ . If the voter  $\kappa$  in  $\mathcal{P}'$  prefers  $c_i$  over  $c_{i+1}$  in  $\mathcal{P}'$ , then the oracle fixes the preference  $\succ_\kappa$  to be  $c_1 \succ c_2 \succ \dots \succ c_{i-1} \succ c_{i+1} \succ c_i \succ c_{i+2} \succ \dots \succ c_m$ ; otherwise the oracle fixes  $\succ_\kappa$  to be  $\mathcal{Q}$ . The algorithm fails to correctly output the preference of the voter  $\kappa$  in both the cases. Also the final profile with the oracle is single crossing with respect to the tree  $\mathcal{T}$ . Hence,  $\mathcal{A}$  must compare  $c_i$  and  $c_{i+1}$  for every voter corresponding to every leaf node of  $\mathcal{T}$  and for every odd integer  $i \in [m]$  and thus has query complexity  $\Omega(mn)$ .  $\square$



Hence, the query complexity of PREFERENCE ELICITATION for the weakly single crossing profiles on trees, does not depend substantially on how the preferences are accessed and whether we know a single crossing tree of the single crossing closure of the input profile. This is in sharp contrast to the corresponding results for single crossing domain where the query complexity for PREFERENCE ELICITATION improves substantially if we know a single crossing ordering and we have a random access to the preferences than if we only have sequential access to preferences [DM16b].

## 5 Conclusion and Future Work

We have shown that, for every weakly single crossing profile on trees, there always exists a unique inclusion-minimal single crossing profile on trees containing it. We exploit this uniqueness property along with the known polynomial time recognition algorithm for single crossing profiles on trees to design a polynomial time recognition algorithm for weakly single crossing profiles on trees. We have also shown that the query complexity for preference elicitation for weakly single crossing profiles on trees is small – the query complexity bounds are similar to the corresponding bounds for single peaked [Con09] and single crossing profiles [DM16b]. Moreover, we have proved that our preference elicitation algorithm makes an optimal number of queries up to constant factors for a large number of voters. Hence, our results in this paper, along with the fact that the weakly single crossing domain on trees inherits most of the desirable properties from the single crossing domain on trees, shows that the weakly single crossing domain on trees is an important generalization of the single crossing domain on trees. An immediate future work is to extend our recognition and elicitation algorithms to the more general median graphs. Characterizing weakly single crossing profiles in trees in terms of forbidden structures is another important direction of research.

## Acknowledgement:

Palash Dey gratefully acknowledges all the useful discussions with Neeldhara Misra during the course of the work. Palash Dey also thanks Satyanath Bhat for proposing the terminology “weakly single crossing profiles on trees” and providing useful inputs to make some of the proofs clearer. Palash Dey also wishes to gratefully acknowledge support from Google India for providing him with a special fellowship for carrying out his doctoral work.

## References

- [BBHH15] Felix Brandt, Markus Brill, Edith Hemaspaandra, and Lane A Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *J. Artif. Intell. Res.*, pages 439–496, 2015.

- [BC84] Douglas H Blair and David L Crawford. Labor union objectives and collective bargaining. *Q. J. Econ.*, pages 547–566, 1984.
- [BCW13] Robert Brederick, Jiehua Chen, and Gerhard J Woeginger. A characterization of the single-crossing domain. *Soc. Choice Welf.*, 41(4):989–998, 2013.
- [BCW16] Robert Brederick, Jiehua Chen, and Gerhard J. Woeginger. Are there any nicely structured preference profiles nearby? *Math. Soc. Sci.*, 79:61–73, 2016.
- [Bla48] Duncan Black. On the rationale of group decision-making. *J. Polit. Econ.*, pages 23–34, 1948.
- [BM11] Salvador Barberà and Bernardo Moreno. Top monotonicity: A common root for single peakedness, single crossing and the median voter result. *GEB*, 73(2):345–359, 2011.
- [BNM<sup>+</sup>58] Duncan Black, Robert Albert Newing, Iain McLean, Alistair McMillan, and Burt L Monroe. *The theory of committees and elections*. Springer, 1958.
- [BSU13] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the computation of fully proportional representation. *J. Artif. Intell. Res.*, 47:475–519, 2013.
- [CGS12] Denis Cornaz, Lucie Galand, and Olivier Spanjaard. Bounded single-peaked width and proportional representation. In *Proc. Twentieth European Conference on Artificial Intelligence (ECAI)*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 270–275. IOS Press, 2012.
- [Con09] Vincent Conitzer. Eliciting single-peaked preferences using comparison queries. *J. Artif. Intell. Res.*, 35:161–191, 2009.
- [CPS14] Adam Clearwater, Clemens Puppe, and Arkadii Slinko. The single-crossing property on a tree. *arXiv preprint arXiv:1410.2272*, 2014.
- [CPS15] Adam Clearwater, Clemens Puppe, and Arkadii Slinko. Generalizing the single-crossing property on lines and trees to intermediate preferences on median graphs. In *Proc. 24th International Conference on Artificial Intelligence, IJCAI*, pages 32–38. AAAI Press, 2015.
- [CSS99] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *J. Artif. Int. Res.*, 10(1):243–270, May 1999.
- [Dem82] Gabrielle Demange. Single-peaked orders on a tree. *Math. Soc. Sci.*, 3(4):389–396, 1982.
- [Dem12] Gabrielle Demange. Majority relation and median representative ordering. *SERIEs*, 3(1-2):95–109, 2012.

- [Die97] Reinhard Diestel. *Graph theory*. Graduate texts in mathematics. Springer, New York, Berlin, Paris, 1997. Trad. de : Graphentheorie, paru en 1996 chez Springer.
- [DKNS01] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. 10th International Conference on World Wide Web*, WWW '01, pages 613–622, New York, NY, USA, 2001. ACM.
- [DM16a] Palash Dey and Neeldhara Misra. Elicitation for preferences single peaked on trees. In *Proc. Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 215–221, 2016.
- [DM16b] Palash Dey and Neeldhara Misra. Preference elicitation for single crossing domain. In *Proc. Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 222–228, 2016.
- [Dod76] Charles Lutwidge Dodgson. *A method of taking votes on more than two issues*. 1876.
- [DS74] Peter A Diamond and Joseph E Stiglitz. Increases in risk and in risk aversion. *J. Econ. Theory*, 8(3):337–360, 1974.
- [EF14] Edith Elkind and Piotr Faliszewski. Recognizing 1-euclidean preferences: An alternative approach. In *Algorithmic Game Theory - 7th International Symposium, SAGT 2014, Haifa, Israel, September 30 - October 2, 2014. Proceedings*, pages 146–157, 2014.
- [EFLO15a] Edith Elkind, Piotr Faliszewski, Martin Lackner, and Svetlana Obraztsova. The complexity of recognizing incomplete single-crossing preferences. In *Proc. Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 865–871, 2015.
- [EFLO15b] Edith Elkind, Piotr Faliszewski, Martin Lackner, and Svetlana Obraztsova. The complexity of recognizing incomplete single-crossing preferences. In *Proc. Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 865–871, 2015.
- [EFS14] Edith Elkind, Piotr Faliszewski, and Piotr Skowron. A characterization of the single-peaked single-crossing domain. In *Proc. Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 654–660, 2014.
- [EL14] Edith Elkind and Martin Lackner. On detecting nearly structured preference profiles. In *Proc. Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 661–667, 2014.

- [EL15] Edith Elkind and Martin Lackner. Structure in dichotomous preferences. In *Proc. Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2019–2025, 2015.
- [ELP13] Gábor Erdélyi, Martin Lackner, and Andreas Pfandler. Computational aspects of nearly single-peaked electorates. In *Proc. Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*, 2013.
- [ELP16] Edith Elkind, Martin Lackner, and Dominik Peters. Preference restrictions in computational social choice: Recent progress. In *Proc. Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4062–4065, 2016.
- [FHH14] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. *Artif. Intell.*, 207:69–99, February 2014.
- [HM97] Melvin J Hinich and Michael C Munger. *Analytical politics*. Cambridge University Press, 1997.
- [JSA08] Benjamin G. Jackson, Patrick S. Schnable, and Srinivas Aluru. Consensus genetic maps as median orders from inconsistent sources. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 5(2):161–171, 2008.
- [Kem59] John G Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.
- [Kno10] Vicki Knoblauch. Recognizing one-dimensional euclidean preference profiles. *J. Math. Econ.*, 46(1):1–5, 2010.
- [Kun15] Fan-Chin Kung. Sorting out single-crossing preferences on networks. *Soc. Choice Welf.*, 44(3):663–672, 2015.
- [Lac14] Martin Lackner. Incomplete preferences in single-peaked electorates. In *Proc. Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, pages 742–748, 2014.
- [Lev75] Arthur Levenglick. Fair and reasonable election systems. *Behavioral Science*, 20(1):34–46, 1975.
- [MG04] Vincent Merlin and Wulf Gaertner. Domain conditions in social choice theory, 2004.
- [Mir71] James A Mirrlees. An exploration in the theory of optimum income taxation. *Rev. Econ. Stud.*, pages 175–208, 1971.

- [Mou91] Hervi Moulin. *Axioms of cooperative decision making*. Number 15. Cambridge University Press, 1991.
- [MR81] Allan H Meltzer and Scott F Richard. A rational theory of the size of government. *J. Polit. Econ.*, pages 914–927, 1981.
- [Rob77] Kevin WS Roberts. Voting over income tax schedules. *J Public Econ*, 8(3):329–340, 1977.
- [SYFE13] Piotr Skowron, Lan Yu, Piotr Faliszewski, and Edith Elkind. The complexity of fully proportional representation for single-crossing electorates. In *International Symposium on Algorithmic Game Theory*, pages 1–12. Springer, 2013.
- [Tri89] Michael A Trick. Recognizing single-peaked preferences on a tree. *Math. Soc. Sci.*, 17(3):329–334, 1989.